**Student name:** Federico Watkins

**CSIS 3810:** Operating Systems

**Chapter:** 10-11 File Systems Structure & Interface

**Assignment:** 8

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. 🡪 10.11 | 1. 🡪 10.12 | 1. 🡪 11.10 | 1. 🡪 11.13 | 1. 11.16 |

1. Suppose that a disk drive has 5,000 cylinders, numbered 0 to 4999. The drive is currently serving a request at cylinder 2150, and the previous request was at cylinder 1805. The queue of pending requests, in FIFO order, is:

2069, 1212, 2296, 2800, 544, 1618, 356, 1523, 4965, 3681

Starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests for each of the following disk-scheduling algorithms?

1. FCFS

= |2150-2069| + |2069-1212| + |1212-2296| + |2296-2800| + |2800-544| + |544-1618| +

|1618-356| + |356-1523| + |1523-4965| + |4965-3681|

= 81 + 857 + 1084 + 504 + 2256 + 1074 + 1262 + 1167 + 3442 + 1284

= **13011**

1. SSTF

= |2150-2069| + |2069-2296| + |2296-2800| + |2800-3681| + |3681-4965| +

|4965-1618| + |1618-1523| + |1523-1212| + |1212-544| + |544-365|

= 81 + 227 + 504 + 881 + 1284 + 3347 + 95 + 311 + 668 + 188

= **7586**

1. SCAN

= |2150-2069| + |2069-1618| + |1618-1523| + |1523-1212| + |1212-544| +

|544-356| + |356-0| + |0-2296| + |2296-2800| + |2800-3681| + |3681-4965|

= 81 + 451 + 95 + 311 + 668 + 188 + 356 + 2296 + 504 + 881 + 1275

= **7106**

1. LOOK

= |2150-2069| + |2069-1618| + |1618-1523| + |1523-1212| + |1212-544| + |544-356| +

|356-2296| + |2296-2800| + |2800-3681| + |3681-4965|

= 81 + 451 + 95 + 311 + 668 + 188 + 1940 + 504 + 881 + 1284

= **6403**

1. C-SCAN

= |2150-2296| + |2296-2800| + |2800-3681| + |3681-4965| + |4965-4999| + |4999-0| +

|0-356| + |356-544| + |544-1212| + |1212-1523| + |1523-1618| + |1618-2069|

= 146 + 504 + 881 + 1284 +34 + 4999 +356 + 188 + 668 + 311 + 95 + 451

= **9917**

1. C-LOOK

= |2150-2069| + |2069-1618| + |1618-1523| + |1523-1212| + |1212-544| + |544-356| +

|356-4965| + |4965-3681| + |3681-2800| + |2800-2296|

= 81 + 451 + 95 + 311 + 668 + 188 + 4609 + 1284 + 881 + 504

= **9072**

1. Elementary physics states that when an object is subjected to a constant acceleration *a,* the relationship between distance *d* and time *t* is given by. Suppose that, during a seek, the disk in Exercise 10.11 accelerates the disk arm at a constant rate for the first half of the seek, then decelerates the disk arm at the same rate for the second half of the seek. Assume that the disk can perform a seek to an adjacent cylinder in 1 millisecond and a full-stroke seek over all 5,000 cylinders in 18 milliseconds.
2. The distance of a seek is the number of cylinders over which the head moves. Explain why the seek time is proportional to the square root of the seek distance.

Rearranging for *t* shows that

1. Write an equation for the seek time as a function of the seek distance. This equation should be of the form, where *t* is the time in milliseconds and *L* is the seek distance in cylinders.

Solving the following equations with (*t* = 1*, L* = 1) and then (*t* = 18*, L* = 4999) gives

.

1. Calculate the total seek time for each of the schedules in Exercise 10.11. Determine which schedule is the fastest (has the smallest total seek time).

Total seek times:

FCFS = 65.20

SSTF = 31.52

SCAN = 62.02

LOOK = 40.29

C-SCAN = 62.10

C-LOOK 40.42

* **SSTF is fastest here**.

1. The percentage speedupis the time saved divided by the original time. What is the percentage speedup of the fastest schedule over FCFS?

.

Hence, with respect to the seek time, the percentage speedup of SSTF over FCFS is **52%**.

1. The open-file table is used to maintain information about files that are currently open. Should the operating system maintain a separate table for each user or just maintain one table that contains references to files that are being accessed by all users at the current time? If the same file is being accessed by two different programs or users, should there be separate entries in the open file table?

Keeping a central open-file table allows the operating system to perform an otherwise infeasible following operation. For example, a file that is currently being accessed by one or more processes. If the file is deleted, then it shouldn’t be removed from the disk until all processes accessing the file have closed it. Such checks can be performed only if there is centralized accounting of number of processes accessing the file. On the other hand, if two processes are accessing the file, then two separate states need to be maintained to keep track of the current location of which parts of the file are being accessed by the two processes. This requires the operating system to maintain separate entries for the two processes.

1. Some systems automatically open a file when it is referenced for the first time, and close the file when the job terminates. Discuss the advantages and disadvantages of this scheme as compared to the more traditional one, where the user has to open and close the file explicitly.

|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| It relieves the user from the invocation of such functions making it more convenient to the user. | It requires more overhead than the case where explicit opening and closing is required. |

1. Discuss the advantages and disadvantages of supporting links to files that cross mount points (that is, the file link refers to a file that is stored in a different volume).

|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| There is greater transparency in the sense that the user does not need to be aware of mount points and create links in all scenarios. | The file system containing the link might be mounted while the file system containing the target file might not be, and thus transparent access cannot be provided to the file in such a scenario.  The error condition would expose to the user that a link is a dead link and that the link does indeed cross file system boundaries. |

References:

1. Operating System Concepts, Ninth Edition, Abraham Silberschartz, Peter bear Galvin, Greg Gagne. Entire Chapter 10-11 for each question in its respective section.
2. Dr. Szabo’s tutorial: File\_Management(1).ppt and

I learned to describe the physical structure of secondary storage devices and its effects on the uses of the devices. I feel better prepared to explain the performance characteristics of mass-storage devices, evaluate disk scheduling algorithms and discuss operating-system services provided for mass storage, including RAID. I was exposed to the functions of file systems and learned to describe the interfaces to file systems. Also I feel comfortable enough to discuss file-system design tradeoffs, including access methods, file sharing, file locking, and directory structures. Took a closer look at file-system protection. Now I have an improved overall comprehension the operating system’s structure and interface.